

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平8-83257

(43)公開日 平成8年(1996)3月26日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/16	3 8 0 Z			
	4 3 0 B			

審査請求 未請求 請求項の数3 O L (全 7 頁)

(21)出願番号 特願平6-218396

(22)出願日 平成6年(1994)9月13日

(71)出願人 000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地

(71)出願人 391002409
日立システムエンジニアリング株式会社
東京都大田区大森北3丁目2番16号

(72)発明者 柴田 恭徳
東京都大田区大森三丁目2番16号 日立システムエンジニアリング株式会社内

(72)発明者 田中 夏朗
神奈川県秦野市堀山下1番地 株式会社日立製作所汎用コンピュータ事業部内

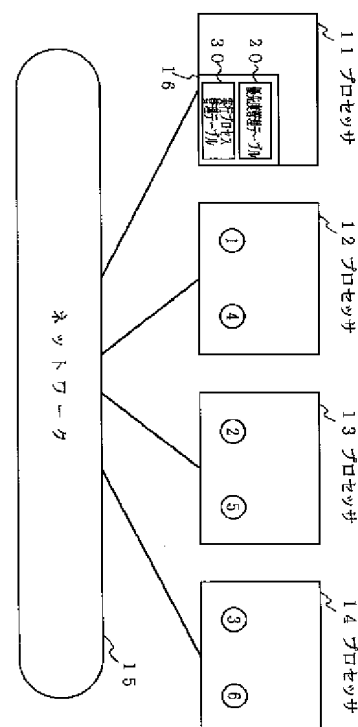
(74)代理人 弁理士 鈴木 誠

(54)【発明の名称】 並列コンピュータシステムのプロセス実行最適化方法

(57)【要約】

【目的】 実行途中で動的に、利用率の高いプロセッサのプロセスを利用率の低いプロセッサへ移動させて、効率良く業務を処理する。

【構成】 管理用プロセッサ11と複数の処理用プロセッサ12, 13, 14がネットワーク15により接続されている。管理用プロセッサ11のメモリ16には、プロセスを移動する際の優先順位を定義した優先度管理テーブル20、各処理用プロセッサの実行プロセス、実行利用率を管理する実行プロセス管理テーブル30がある。管理用プロセッサ11は、実行プロセス管理テーブル30により、利用率の高い処理用プロセッサと利用率の低い処理用プロセッサが存在する場合、該利用率の高いプロセッサで実行中の複数のプロセスの中から、優先度管理テーブル20を参照して優先順位が一番低いプロセスを選択し、利用率の低いプロセッサへ移動する。



1

【特許請求の範囲】

【請求項1】 複数のプロセスを並列に処理する複数の処理用プロセッサと、各処理用プロセッサの実行プロセスを管理する管理用プロセッサと、各プロセッサを接続するネットワークからなる並列コンピュータシステムにおけるプロセス実行最適化方法であって、前記管理用プロセッサは、各処理用プロセッサで処理されるプロセスの優先順位と、各処理用プロセッサの実行利用率を保持し、実行利用率の高い処理用プロセッサで実行中の複数のプロセスの中から優先順位の低いプロセスを選択して、実行利用率の低い処理用プロセッサへ移動せしめることを特徴とする並列コンピュータシステムのプロセス実行最適化方法。

【請求項2】 請求項1記載の並列コンピュータシステムのプロセス実行最適化方法において、前記管理用プロセッサは、プロセスの移動先プロセッサを各処理用プロセッサに連絡することを特徴とする並列コンピュータシステムのプロセス実行最適化方法。

【請求項3】 請求項1あるいは2記載の並列コンピュータシステムのプロセス実行最適化方法において、前記複数の処理用プロセッサの中の任意のプロセッサが前記管理用プロセッサを兼用することを特徴とする並列コンピュータシステムのプロセス実行最適化方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、複数のプロセッサが高速ネットワークに接続され、複数業務の複数プロセスを同時に実行する並列コンピュータシステムにおいて、複数のプロセスを各プロセッサで効率よく分散して実行するプロセス最適化方法に関する。

【0002】

【従来の技術】 一般に並列コンピュータシステムにおいて業務を実行している場合、利用率の高い（負荷の集中した）プロセッサと、利用率の低い（負荷の少ない或いはかかっていない）プロセッサが発生する。このような場合、従来のこの種コンピュータシステムでは、オペレータの操作なしに自動的に性能が足りないプロセッサで実行中のプロセスを、他の性能的に余裕のあるプロセッサへ移動させて行う処理ができなかった。

【0003】

【発明が解決しようとする課題】 従来技術においては、並列コンピュータシステムの各プロセッサで実行するプロセスの振り分けを変更する場合、一々、オペレータの操作を必要とするため、利用率の高いプロセッサと利用率の低いプロセッサがある場合、実行途中で、利用率の高いプロセッサで実行中のプロセスの一部をダイナミックに利用率の低いプロセッサに移動して効率よく処理を行うことができないという問題があった。

【0004】 本発明の目的は、並列コンピュータシステムにおいて、利用率の高いプロセッサのプロセスを、利

2

用率の低いプロセッサへ、実行途中に、オペレータの操作なしに自動的に且つダイナミックに移動して効率よく処理を行う、プロセス実行最適化方法を提供することにある。

【0005】

【課題を解決するための手段】 本発明は、複数プロセスを並列に処理する複数の処理用プロセッサと、これら処理用プロセッサの実行プロセスを管理する管理用プロセッサを具備し、該管理用プロセッサは、各処理用プロセッサで処理されるプロセスの優先順位と、各処理用プロセッサの実行利用率を保持し、自動的に、実行利用率の高い処理用プロセッサで実行中の複数のプロセスの中から優先順位の低いプロセスを選択して、実行利用率の低い処理用プロセッサへ移動するようにしたことである。

【0006】

【作用】 管理用プロセッサは各処理用プロセッサから、一定時間毎あるいは必要に応じ、その時点の実行利用率を収集する。そして、この収集した各処理用プロセッサの実行利用率によって、実行利用率の高い（負荷の集中した）プロセッサで実行中の複数プロセスの中から、優先順位の一番低いプロセスの実行を停止し、該停止までの実行結果、プログラム停止ポイントを当該プロセスと共に、実行利用率の低い（負荷の少ない）プロセッサへ転送する。該実行利用率の低いプロセッサは、転送されたプロセスを、その停止までの実行結果、プログラム停止ポイントに基づいて処理を続行する。なお、管理用プロセッサは、複数の処理用プロセッサの中の任意のプロセッサが兼ねてもよい。

【0007】

【実施例】 以下、本発明の一実施例を図面により説明する。

【0008】 図1は本発明の一実施例のシステム構成図で、プロセッサ11～14が高速なネットワーク15に接続されている。本実施例では、プロセッサ11を管理用プロセッサ、プロセッサ12～14をプロセス実行の処理用プロセッサとし、管理用プロセッサ11のメモリ16に、各処理用プロセッサで実行されるプロセスの優先度を管理する優先度管理テーブル20と、各処理用プロセッサの実行プロセス、規定利用率、実行利用率を管理する実行プロセス管理テーブル30を用意する。①～⑥は、処理用プロセッサ12～14上の現実行中のプロセスを示している。

【0009】 なお、プロセッサ11～14を全て処理用プロセッサとし、その任意のプロセッサが管理用プロセッサを兼ねるようにしてもよい。これは管理用プロセッサが故障した場合に、その機能を他のプロセッサに引き継げる利点がある。

【0010】 図2は、管理用プロセッサ11のメモリ16上にある優先度管理テーブル20の構成例を示したもので、21は実行業務名、22は実行業務の優先順位、

3

23は各実行業務に対応するプロセスID、24はプロセスの優先順位である。本例の場合、業務の優先順位はA>B>Cであり、プロセスの優先順位は各々①>④、②>⑤、③>⑥である。優先順位をつける基準は、業務やプロセスの重要度でもよいし、緊急度でも構わなく、使うシステムにより可変でもよい。

【0011】図3は、管理プロセッサ11の同じくメモリ16上にある実行プロセス管理テーブル30の構成例を示したもので、31はプロセッサ番号、32は各プロセッサでの現実行中のプロセス（実行プロセス）、33は各プロセッサについて実行プロセスを軽減するプロセスを決める目安となる負荷の上限を定めた規定利用率（MAX）、34は実行プロセスの移動先のプロセッサを決める目安となる負荷の下限を定めた規定利用率（MIN）、35は各プロセッサの実際の負荷の利用率（実行利用率）である。実行利用率は、例えば、一定時間当りのメモリアクセス回数（アクセス頻度）で定義する。

【0012】本実施例では、図3のプロセッサ番号1～3は、図1の処理用プロセッサ12、13、14に対応するとする。図3より、プロセッサ12はプロセス①と④、プロセッサ13はプロセス②と⑤、プロセッサ14はプロセス③と⑥を実行し、その実行利用率35はそれぞれ90%、10%、30%である。一方、規定利用率33、34は、ここではプロセッサ12、13、14とも一律に80%、20%としている。

【0013】管理用プロセッサ11は、実行プロセス管理テーブル30を参照して、実行利用率80%以上（負荷が集中している）の処理用プロセッサがあり、しかも、実行利用率20%以下（負荷が少ない）の処理用プロセッサがある場合、実行利用率80%以上の処理用プロセッサで処理している複数のプロセスの中から、優先度管理テーブル20を参照して、優先順位の一番低いプロセスを選択し、その処理を実行利用率20%以下の処理用プロセッサへ移動する。図2および図3においては、プロセッサ番号1の処理用プロセッサ12のプロセス④の処理が、プロセッサ番号2の処理用プロセッサ13に移動せしめられる。

【0014】図4は、各処理プロセッサへのプロセスの配布から、プロセッサの実行利用率によってプロセスを移動させる処理の流れを示すフローチャート、図5はその際に管理用プロセッサと処理用プロセッサ間で授受されるデータ形式を示す図である。以下、図4および図5を参照して、実行利用率の高いプロセッサの実行中プロセスを実行利用率の低いプロセッサに、オペレータの操作なしに自動的に移動させる処理を、具体的に説明する。

【0015】管理用プロセッサ11は、実行業務をスケールリングし、各処理用プロセッサ12、13、14へ実行すべきプロセスを配布する（ステップ401）。図5の501は、この実行プロセス配布時のデータ形式を示

4

したものである。実行プロセスの配布はプロセッサ毎、ブロードキャストのいずれでもよいが、各プロセッサには、プロセス間の通信の必要性などのため、他のプロセッサの実行プロセスIDも配布しておく。また、管理用プロセッサ11は、この実行プロセス配布時、メモリ16上の優先度管理テーブル20に業務毎、プロセス毎の優先度を設定し、実行プロセス管理テーブル30に各管理用プロセッサ12、13、14が実行するプロセスを格納しておく。実行プロセス管理テーブル30の規定利用率33、34はあらかじめ設定しておくか、あるいは、当該ステップ401で実行プロセスと対に設定することでもよい。各プロセッサの実行利用率35は、初期状態ではすべて零である。

【0016】各処理用プロセッサ12、13、14は、管理用プロセッサ11から配布されたプロセスの処理を開始する（ステップ402）。本実施例では、プロセッサ番号1のプロセッサ12はプロセス①と④、プロセッサ番号2のプロセッサ13はプロセス②と⑤、プロセッサ番号3のプロセッサ14はプロセス③と⑥の処理を開始する。

【0017】各処理用プロセッサ12、13、14は、各種データ等を格納する記憶装置を具備しており、該記憶装置をアクセスすることでプロセスを処理し、また、必要があれば他のプロセッサのプロセスと通信して、該他プロセッサの記憶装置をアクセスする。ここでは、この記憶装置に対するアクセス頻度をプロセッサの実行利用率（負荷の集中度）の目安とする。各処理用プロセッサ12、13、14では、プロセスの処理を開始すると、時々刻々記憶装置に対する一定時間当りのアクセス回数をカウントし、その最新の値を実行利用率として保持しておく。なお、実行利用率は、一定時間当りの命令処理数などを目安とすることでもよい。

【0018】管理用プロセッサ11は、各処理用プロセッサ12、13、14の上記実行利用率を収集し、実行プロセス管理テーブル30の実行利用率35を更新する（ステップ403）。実行利用率の収集は、ここでは、管理用プロセッサ11が一定時間毎に各処理用プロセッサ12、13、14に要求を出し、それに応答して、各処理用プロセッサ12、13、14がその時点の実行利用率を管理用プロセッサ11に連絡することで行うとする。図5の502～504は、各処理用プロセッサ12、13、14が実行利用率を管理用プロセッサ11へ連絡する際のデータ形式を示したものである。

【0019】各処理用プロセッサ12、13、14の実行利用率を収集すると、管理用プロセッサ11は、実行プロセス管理テーブル30内の規定利用率（MAX）33と実行利用率35を比較して、実行利用率が80%以上のプロセッサがあるか調べる（ステップ404）。ここで、実行利用率が80%以上のプロセッサがない場合、各処理用プロセッサ12、13、14はいずれも負

5

荷の集中がないと見なし、この場合は何もしない。即ち、各処理用プロセッサ12, 13, 14は、そのまま処理を続行する。

【0020】一方、実行利用率が80%以上のプロセッサがある場合、該プロセッサに負荷が集中していると見なす。この場合、管理用プロセッサ11は、実行プロセス管理テーブル30内の規定利用率(MIN)34と実行利用率35を比較して、実行利用率が20%以下のプロセッサがあるか調べる(ステップ405)。即ち、負荷が集中しているプロセッサのプロセスを移動させる先の、負荷の少ないプロセッサを探す。ここでも、実行利用率が20%以下(負荷が少ない)のプロセッサがない場合、何もしない。この場合、各処理用プロセッサ12, 13, 14はそのまま処理を続行するため、負荷の集中しているプロセッサは、該負荷集中の状態でプロセス処理を続けることとなる。即ち、実行利用率80%以上(負荷集中)のプロセッサはあるが、利用率20%以下(負荷が少ない)のプロセッサがない場合、プロセスを移動させても、移動先プロセッサで実行利用率が80%以上になる可能性が高く、結局、処理効率は悪くなるため、プロセスを移動させずにそのまま処理を続行させるのである。

【0021】実行利用率80%以上(負荷集中)のプロセッサが存在し、かつ、実行利用率20%以下(負荷が少ない)のプロセッサも存在する場合、管理用プロセッサ11は、まず、実行利用率80%以上のプロセッサの実行プロセスの中から、優先度管理テーブル20を参照して、優先順位の一番低いプロセスを選択し、該実行利用率80%以上のプロセッサに当該プロセスの処理を停止させ、該停止した時点までの実行結果、プログラム停止ポイントの収集を行う(ステップ406)。次に、管理用プロセッサ11は、実行利用率20%以下のプロセッサに対して、上記実行利用率80%以上のプロセッサで処理を停止させたプロセス内容、及び、それまでの実行結果、プログラム停止ポイントを転送する(ステップ407)。

【0022】図2、図3において、プロセッサ番号1の処理用プロセッサ12の実行利用率は90%、プロセッサ番号2の処理用プロセッサ13の実行利用率は10%であり、さらに、処理用プロセッサ12で実行中のプロセス①、④の優先順位は①>④である。そこで、管理用プロセッサ11は、実行利用率90%の処理用プロセッサ12に対して、プロセス④の実行を停止させて、該停止した時点までの実行結果、プログラム停止ポイントの収集を行い、それが完了すると、実行利用率10%の処理用プロセッサ13に対して、プロセス②、及び、その停止した時点までのデータ(実行結果、プログラム停止ポイント)を転送する。図5の505は、処理用プロセッサ12にプロセス停止を連絡する際のデータ形式、506は処理用プロセッサ12から停止させたプロセスの

6

データを収集する際のデータ形式、507はプロセス移動先の処理用プロセッサ13へ所望データを転送する際のデータ形式を示したものである。

【0023】実行利用率20%(負荷が少ない)の処理用プロセッサへプロセス及び所望データを転送した後、管理用プロセッサ11は、実行プロセス管理テーブル30の実行プロセス欄32の書き替え、各処理用プロセッサ12, 13, 14に対するプロセスの移動先プロセッサの通知(アドレス変更通知)、及び、プロセス移動先の処理用プロセッサに対するプロセス開始指示を行う(ステップ408)。ここでは、図3の実行プロセス管理テーブル30において、プロセッサ番号1(プロセッサ12)の実行プロセスからプロセス④を削除し、プロセッサ番号2(プロセッサ13)の実行プロセスに該プロセス④を追加する。また、プロセス④の処理用プロセッサがプロセッサ番号2(プロセッサ13)に変更されたことを各処理用プロセッサに通知し、プロセッサ番号2のプロセッサ13にプロセス④の処理開始を指示する。図5の508は、管理用プロセッサ11がプロセスの処理用プロセッサの変更(アドレス変更)を各処理用プロセッサ12, 13, 14に対してブロードキャストなどで連絡する際のデータ形式、509は管理用プロセッサ11が処理用プロセッサ13に対してプロセス④の開始指示を行う際のデータ形式を示したものである。

【0024】プロセスの開始指示を受け取った処理用プロセッサは、該プロセスの処理を再開する(ステップ409)。ここでは、プロセッサ番号2の処理用プロセッサ13が、プロセス④について、転送されたプログラム停止ポイントから処理を再開することとなる。

【0025】一般に、実行利用率80%以上の高負荷プロセッサの数をM、実行利用率20%以下の低負荷プロセッサの数をNとした場合、 $M \leq N$ の場合は、高負荷プロセッサは実行利用率の高いものから順に、低負荷プロセッサは実行利用率の低いものから順に、それぞれプロセスの移動元/移動先プロセッサを対応づけて、対象とする優先順位の低いプロセスを移動すればよい。また、 $M > N$ の場合は、M個の高負荷プロセッサのうち、実行利用率の高い順にN個のプロセッサを選択し、それらを上記 $M \leq N$ の場合と同様にして、N個の低負荷プロセッサと対応づけて、対象とするプロセスを移動すればよい。

【0026】図4のステップ404~409の処理は、各処理用プロセッサ12, 13, 14の実行利用率を収集する度に繰り返される。これにより、実行利用率の高い(負荷の集中している)プロセッサのプロセスを、実行利用率の低い(負荷の少ない)プロセッサへ動的(ダイナミック)に振り分け、処理を続行することが可能になる。この振り分けの際、移動対象プロセスの処理の一時停止が伴うが、該プロセスは、実行利用率の高いプロセッサで実行中のマルチプロセスの中から優先順位の一

7

番低いプロセスが選択されるため、影響を最小限に抑えることができる。

【0027】以上、実施例の説明では実行利用率の収集を、管理用プロセッサが一定時間毎に各処理用プロセッサに要求を出し、それに応答して、各処理用プロセッサが実行利用率を管理プロセッサに連絡することで行うとしたが(図5の502, 503, 504)、各処理用プロセッサがそれぞれ不定期(任意)に実行利用率を送信し、それを管理用プロセッサが受取って、その都度、実行プロセス管理テーブルの該当プロセッサ番号に対応する実行利用率35を更新することでもよい。この場合、管理用プロセッサから各処理用プロセッサへ実行利用率の送信要求を発行するシーケンスを省略である。なお、この場合の図4のステップ404~409の処理は、ある処理用プロセッサから実行利用率を収集する度に行ってもよいし、あるいは、それとは無関係に一定時間ごとに行うことでもよい。

【0028】

【発明の効果】本発明による並列コンピュータシステムのプロセス実行最適化方法によれば、次のような効果が得られる。

【0029】(1) 処理用プロセッサの実行利用率によって、実行中のプロセスが自動的かつダイナミックに、利用率の高い(負荷の集中した)プロセッサから利用率の低い(負荷の少ない)プロセッサへ移動させることで、プロセッサを効率良く使って、プロセスを効率よく処理させることが可能となり、実行業務の処理時間の短縮ができる。

【0030】(2) 利用率の高いプロセッサから利用率の低いプロセッサへの移動させるプロセスは、あらかじめ定めた業務の優先度等に依存したプロセスの優先順位に基づいて、実行中の複数のプロセスの中から優先順位の一番低いプロセスが対象となるため、実行する業務への

8

影響を最小限に抑えることができる。

【0031】(3) 管理用プロセッサは、利用率の高いプロセスから利用率の低いプロセッサへプロセスを移動した場合、該プロセスと共にプロセス移動先プロセッサを全処理用プロセッサへ連絡することにより、以後の処理でプロセス間通信の必要が生じた際、処理用プロセッサは管理用プロセッサの仲なしに該プロセス間通信を行うことができる。

【0032】(4) 複数の処理用プロセッサの中の任意のプロセッサが管理用プロセッサを兼用する構成とすることにより、管理用プロセッサが故障した場合、その管理情報を他のプロセッサに転送することで、該プロセッサを管理用プロセッサとしてシステムを継続して移動することが可能になる。

【図面の簡単な説明】

【図1】本発明による並列コンピュータシステムの一実施例のブロック図である。

【図2】管理用プロセッサ上の優先度管理テーブルの構成例を示す図である。

【図3】管理用プロセッサ上の実行プロセス管理テーブルの構成例を示す図である。

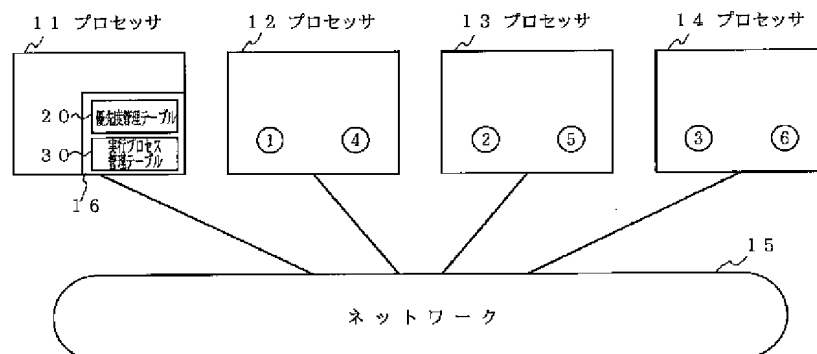
【図4】本発明によるプロセス実行最適化方法の一実施例の処理フロー図である。

【図5】管理用プロセッサと処理用プロセッサとの間で授受されるデータ形式の一例を示す図である。

【符号の説明】

- 11 管理用プロセッサ
- 12~14 処理用プロセッサ
- 15 ネットワーク
- 16 管理用プロセッサのメモリ
- 20 優先度管理テーブル
- 30 実行プロセス管理テーブル

【図1】



【図2】

優先度管理テーブル 20

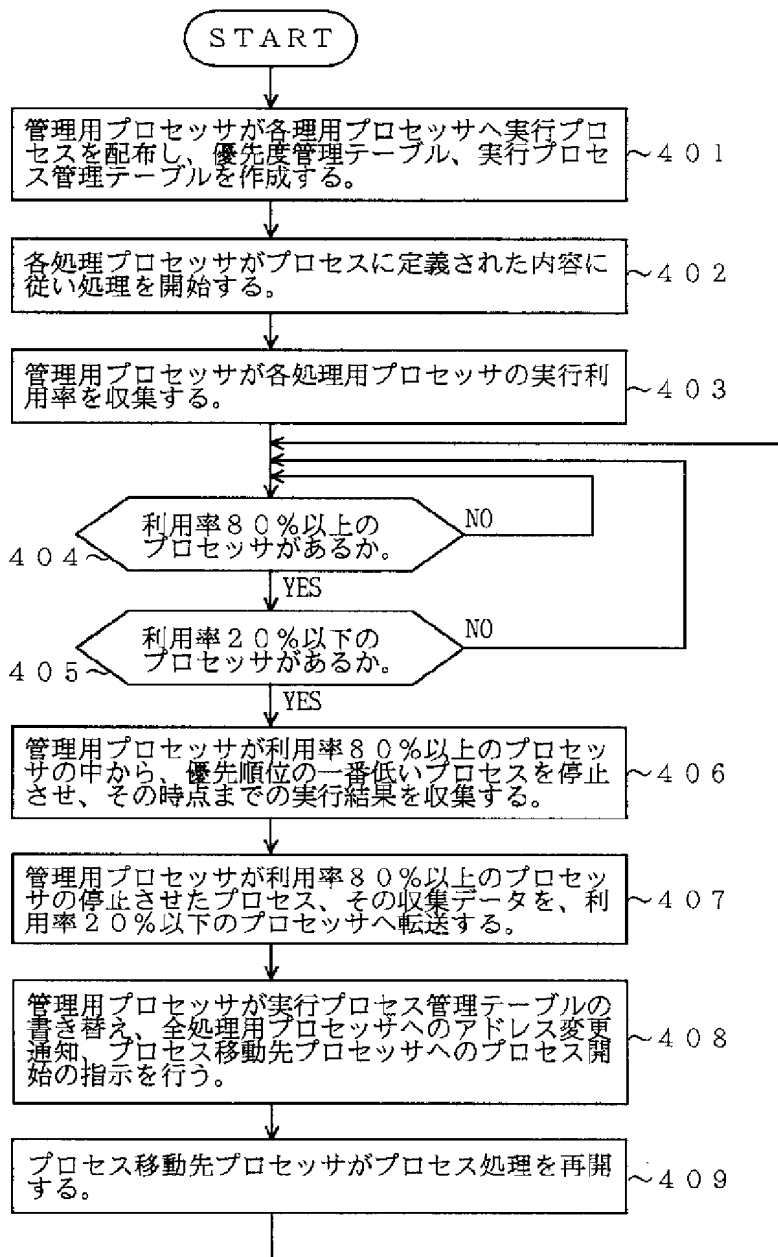
21 業 務	22 優先度	23 プロセス	24 優先度
A	1	①	1
		④	2
B	2	②	1
		⑤	2
C	3	③	1
		⑥	2

【図3】

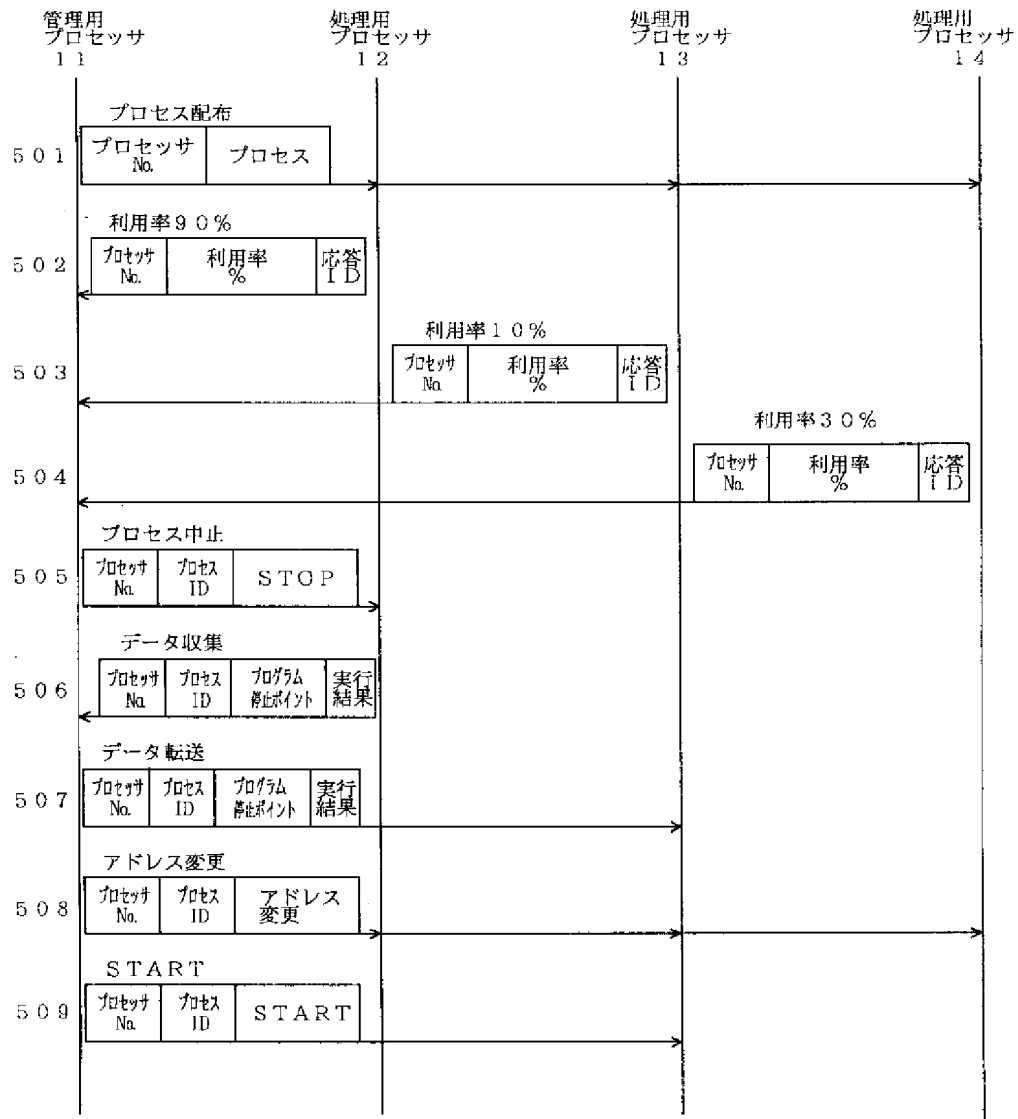
実行プロセス管理テーブル 30

31 プロセス番号	32 実行プロセス	33 規定利用率(MAX)	34 規定利用率(MIN)	35 実行利用率
1	① ④	80%	20%	90%
2	② ⑤	80%	20%	10%
3	③ ⑥	80%	20%	30%

【図4】



【図5】



DERWENT-ACC-NO: 1996-218793

DERWENT-WEEK: 199622

COPYRIGHT 2009 DERWENT INFORMATION LTD

TITLE: Process execution optimisation method
for parallel computer system by
sending process to processor with low
utilisation factor when processor with
high utilisation factor is executing
many processes

INVENTOR: SHIBATA T; TANAKA N

PATENT-ASSIGNEE: HITACHI LTD[HITA] , HITACHI
SYSTEM ENG KK[HITAN]

PRIORITY-DATA: 1994JP-218396 (September 13, 1994)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE
JP 08083257 A	March 26, 1996	JA

APPLICATION-DATA:

PUB-NO	APPL- DESCRIPTOR	APPL- NO	APPL- DATE
JP 08083257A	N/A	1994JP- 218396	September 13, 1994

INT-CL-CURRENT:

TYPE	IPC DATE
CIPP	G06F15/16 20060101
CIPS	G06F15/177 20060101
CIPS	G06F9/50 20060101

ABSTRACTED-PUB-NO: JP 08083257 A

BASIC-ABSTRACT:

The method involves connecting a process controller (11) and several processor (12-14) to a single network (15). The process controller includes a memory circuit (16). The memory includes a priority control table (20) and an execution process control table (30). The priority control table defines the precedence of a logical operator at the time of moving a process and the execution procedure of the processor for each processing.

The execution process table controls the execution utilisation factor in the memory. A process is sent to a processor with a low utilisation factor according to precedence of logical operator of the priority control table when the processor with a high utilisation factor is processing a lot.

ADVANTAGE - Enables moving of process from processor with high utilisation factor to processor with low utilisation factor in middle of execution. Raises processing distribution efficiency.

CHOSEN-DRAWING: Dwg.1/5

TITLE-TERMS: PROCESS EXECUTE OPTIMUM
 METHOD PARALLEL COMPUTER
 SYSTEM SEND PROCESSOR LOW
 UTILISE FACTOR HIGH

DERWENT-CLASS: T01

EPI-CODES: T01-F02C; T01-M02C;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: 1996-183757